ERP PORTAL

A PROJECT REPORT
For
Major Project – III(CA301P)
Session (2025-26)

Submitted By

Monty Sharma (202410116100124)
Manish Kumar (202410116100113)
Mohd Kasif Khan (202410116100122)

Submitted in partial fulfilment of the
Requirements for the degree of

MASTER OF COMPUTER APPLICATION

Under the supervision of
Dr. Ankit Verma
Associate Professor

Submitted to

DEPARTMENT OF COMPUTER APPLICATIONS
KIET GROUP OF INSTITUTIONS, Ghaziabad
Uttar Pradesh-201206

(Dec-2025)

DECLARATION

We hereby declare that the work presented in this report entitled “ERP PORTAL", was carried out by us. We have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute. We have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. We have used quotation marks to identify verbatim sentences and given credit to the original authors/sources. We affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, We shall be fully responsible and answerable.

Monty Sharma
(202410116100124)
Manish Kumar
(202410116100113)
Mohd Kasif Khan
(202410116100122)

CERTIFICATE

Certified that **Monty Sharma** (202410116100124), **Manish Kumar** (202410116100113), **Mohd Kasif Khan** (202410116100122) has/ have carried out the project work having "ERP Portal" (Major Project-III, CA301P) for Master of Computer Application from Dr. A.P.J. Abdul Kalam Technical University (AKTU) (formerly UPTU), Lucknow under my supervision. The project report embodies original work, and studies are carried out by the student himself/herself and the contents of the project report do not form the basis for the award of any other degree to the candidate or to anybody else from this or any other University/Institution.

Dr. Ankit Verma	**Dr. Sachin Malhotra**
Associate Professor	**Dean**
Department of Computer Applications	**Department of Computer Applications**
KIET Group of Institutions, Ghaziabad	**KIET Group of Institutions, Ghaziabad**

ABSTRACT

The ERP Portal is a comprehensive web-based system designed to digitalize and streamline core administrative, academic, and operational activities within an educational institution. Built using the MERN stack (MongoDB, Express.js, React.js, and Node.js), this platform ensures scalability, high performance, and a seamless user experience. The main goal of the system is to improve efficiency, transparency, and accessibility by providing a centralized portal for managing student information, faculty operations, and institutional workflows.

For students, the portal offers an intuitive and responsive interface to access academic records, attendance history, examination schedules, results, and fee payment details. Faculty members can manage class schedules, record student attendance, upload grades, announce assignments, and communicate with students directly from the portal. The administrative team benefits from centralized access to crucial modules such as human resources, payroll, finance, inventory, and overall institutional reporting, enabling real-time data analytics and informed decision-making.

The application uses React.js for the frontend to deliver a dynamic and interactive user interface, while Node.js and Express.js manage backend APIs, workflow automation, and secure authentication. MongoDB serves as the database to efficiently store, retrieve, and manage structured and unstructured data. Core functionalities include attendance tracking, task and assignment management, payroll processing, leave and fee management, and performance evaluation.

By replacing manual workflows and paperwork with automated processes, the ERP Portal significantly improves operational efficiency, reduces errors, and enhances communication among students, faculty, and administrators. Ultimately, this MERN-based ERP Portal establishes a fully integrated, user-centric digital environment that supports better resource planning, simplifies decision-making, and elevates institutional productivity and service quality

ACKNOWLEDGEMENT

Success in life is never attained single-handedly. I am deeply grateful to my project supervisor, **Dr. Ankit Verma**, for his invaluable guidance, unwavering support, and encouragement throughout my project work. His enlightening ideas, constructive comments, and thoughtful suggestions have greatly contributed to the completion of this project.

I would also like to extend my heartfelt thanks to **Dr. Sachin Malhotra**, Professor and Dean, Department of Computer Applications, for his insightful feedback and administrative support on various occasions, which proved to be immensely helpful during critical stages of the project.

I am fortunate to have many understanding friends who have supported me in numerous ways during challenging moments. Their assistance and companionship have been a constant source of motivation.

Finally, my sincere gratitude goes to my family members and all those who have directly or indirectly provided me with moral support, encouragement, and assistance. Their unwavering belief in me and their continuous efforts to keep my life filled with happiness and joy made the completion of this project possible.

Monty Sharma
Manish Kumar
Mohd Kasif Khan

TABLE OF CONTENT

Page No.

List of Figure

INTRODUCTION

1.1 General

Enterprise Resource Planning (ERP) is an integrated software solution that enables organizations to manage multiple functions and workflows through a unified platform. Rather than using separate tools or disconnected systems for different tasks, an ERP application brings together core processes such as academics, administration, finance, human resources, and resource management into a single centralized environment. This integration ensures that data flows seamlessly across departments, reduces manual effort, and supports efficient decision-making through real-time information access.

In many educational institutions, daily activities such as maintaining student records, monitoring attendance, processing fee payments, scheduling classes, and managing faculty tasks are often handled through manual documentation, spreadsheets, or isolated applications. These disconnected methods frequently lead to delays, data duplication, miscommunication, and lack of transparency. Additionally, retrieving accurate and updated information becomes challenging when data is scattered across multiple systems.

To overcome these limitations, the proposed ERP Portal is developed using the MERN stack—MongoDB, Express.js, React.js, and Node.js. This modern technology stack enables the creation of a fully web-based, scalable, and interactive platform. React.js handles the frontend, offering a dynamic and user-friendly interface, while Node.js and Express.js manage the backend operations and server-side logic. MongoDB provides secure and flexible data storage, allowing the system to efficiently handle large sets of institutional records.

The ERP Portal is designed to meet the operational needs of educational institutions such as colleges and universities. It digitizes and centralizes key academic and administrative processes, enabling students, faculty members, and administrators to interact with the system based on their assigned roles. Students can access attendance records, timetables, results, and fee details; faculty can manage courses, monitor student progress, and handle academic tasks; and administrators can oversee institutional resources, generate reports, and manage workflows.

By integrating all these functions into a single platform, the MERN-based ERP Portal reduces manual work, minimizes paperwork, enhances coordination between departments, and ensures real-time availability of accurate data. This system ultimately improves operational efficiency, supports data-driven decision-making, and helps institutions deliver better academic and administrative services.

Key Features and Benefits of ERP Portal:

The proposed ERP Portal, built using the MERN stack (MongoDB, Express.js, React.js, and Node.js), provides an integrated digital environment that streamlines internal operations and improves data accessibility across an institution. The key features and benefits include:

• Centralized Data Management:
All institutional information—such as student records, faculty data, attendance, fee transactions, payroll, and administrative workflows—is stored in a centralized MongoDB database. This ensures a single source of truth, reduces duplication, and allows seamless data sharing across modules.

• Real-Time Data Access and Updates:
Powered by React.js on the frontend and Node.js/Express.js on the backend, the portal enables users to view, update, or track information instantly. Real-time synchronization ensures faster decision-making and prevents delays caused by outdated records.

• Secure Role-Based Access Control:
The system assigns role-specific access privileges (e.g., student, faculty, administration). Each user can only view and interact with modules relevant to their role, ensuring data privacy, enhanced security, and a streamlined user experience.

• Process Automation and Workflow Efficiency:
Routine tasks such as attendance recording, grade submission, fee reminders, leave approvals, and payroll calculations are automated through backend services. Automation reduces manual intervention, minimizes errors, and saves significant administrative effort.

• High Scalability and Modular Design:
Built using a component-based frontend and API-driven backend architecture, the portal can easily be expanded with new modules (e.g., examination, library, hostel management). MongoDB's flexible schema also supports future data growth and institutional scaling.

1.2 Overview of the ERP Portal

The ERP Portal developed using the MERN stack (MongoDB, Express.js, React.js, and Node.js) is a web-based system designed to digitalize and centralize key administrative and academic operations within an educational institution. Instead of relying on multiple disconnected tools or manual processes, the portal brings all major departments—such as Academic Administration, Human Resources, Finance, Student Services, and Payroll—into a single integrated platform.

Through its modern web interface built with React.js, users can easily navigate system modules and access real-time data based on their assigned roles. Node.js and Express.js manage backend operations by handling API requests, business logic, and workflow automation, while MongoDB securely stores institutional data such as student records, staff information, attendance logs, and transaction details.

By offering a unified system, the ERP Portal eliminates redundancy, improves transparency, and strengthens inter-department collaboration. The system supports faster decision-making through real-time insights and structured data flow, ultimately enhancing operational efficiency and allowing institutions to manage their resources more effectively.

Purpose of the ERP Portal

The purpose of the MERN-based ERP Portal is to provide a centralized, web-based platform that simplifies and automates institutional operations. By integrating multiple functions into a common system, the portal enables users—such as administrators, faculty, and students—to efficiently manage and monitor academic and administrative activities. The system supports key processes including student and faculty management, attendance tracking, assignment and task handling, fee and payroll generation, and resource monitoring.

Through its modern architecture using React.js for the frontend, Node.js and Express.js for backend operations, and MongoDB for secure data storage, the ERP Portal ensures smooth data flow, real-time updates, and improved accessibility. Its primary goal is to minimize manual workload, reduce errors, improve transparency, and enhance overall operational efficiency within the institution.

Key Modules of the ERP Portal

The ERP Portal developed using the MERN stack organizes institutional activities into four primary modules: **Admin Module, Principal Module, Teacher Module, and Coordinator Module.** Each module is designed with role-based accessibility, ensuring that users only interact with features relevant to their responsibilities. This modular approach enhances data security, improves accountability, and enables seamless collaboration among stakeholders.

1. Admin Module

The Admin Module serves as the backbone of the portal. It is responsible for system configuration, user onboarding, and overall maintenance of institutional data.

Key Functionalities:

- **UserAccount Role Management:**
  The admin creates and manages user accounts for Principals, Coordinators, Teachers, and Students. Each user is assigned a specific role that defines their permissions and access within the system.
- **Master Data and Resource Setup:**
  The Admin configures essential data such as department creation, course allocation, academic session setup, and timetable templates. Any new changes or structural updates within the institution are handled through this module.
- **Data Security and Monitoring:**
  This module has the highest level of access control. The Admin can monitor login activities, manage backups, and ensure safe handling of institutional data stored in MongoDB.

The Admin Module ensures smooth functioning of the entire ERP system by acting as a central authority for user and system management.

2. Principal Module

The Principal Module provides complete visibility over academic and administrative activities at the institution level. It focuses on monitoring, reviewing, and decision-making.

Key Functionalities:

- **Institution-Level Monitoring:**
  The Principal can view consolidated dashboards showing key statistics such as daily

attendance percentage, teacher activity reports, task completion status, and department progress summaries.

- **Approval Workflow Management:**
  The Principal handles high-level approvals such as leave applications submitted by teachers, academic schedule approvals, and resource allocation requests.
- **Performance and Reporting:**
  The Principal can review performance reports of teachers, attendance trends of students, and monitor academic progress of departments. Data-driven analytics support informed decision-making.

The Principal Module empowers top-level management with insights, ensuring better oversight and institutional governance.

3. Teacher Module

The Teacher Module is designed to support daily academic responsibilities and promote effective student management.

Key Functionalities:

- **Attendance and Academic Records Management:**
  Teachers can mark attendance, upload grades, schedule assessments, create and assign tasks, and track student submissions.
- **Communication and Interaction:**
  Teachers can send announcements or updates to students, upload learning materials, and respond to queries submitted through the portal.
- **Request Handling:**
  Teachers can submit resource or leave requests, which are forwarded to the Coordinator or Principal depending on the workflow.

This module ensures that teachers can focus more on academic responsibilities and less on repetitive administrative tasks.

4. Coordinator Module

The Coordinator Module acts as a bridge between Teachers and the Principal. It enables task supervision, departmental coordination, and monitoring.

Key Functionalities:

- **Task Assignment and Workflow Allocation:**
  Coordinators assign tasks to teachers (such as preparing lesson plans, updating attendance, document uploads) and monitor progress through status updates and deadlines.
- **Academic Progress Monitoring:**
  Coordinators receive attendance reports, task status updates, and academic submissions from teachers and ensure timely completion before forwarding to the Principal.
- **Request Verification:**
  Any request raised by teachers — such as leave, schedule changes, or departmental needs — is validated by the Coordinator before escalation to higher authority.

Salient Features

• **Accessibility:**
The ERP Portal is a fully web-based application developed using React.js for the frontend and Node.js/Express.js for the backend, making it accessible from any device with an internet connection. Users—whether Admin, Principal, Teacher, or Coordinator—can securely log in and manage operations from any location.

• **Security and Role-Based Authorization:**
The system incorporates secure authentication using JWT (JSON Web Tokens) and role-based access control. Each user is granted access only to the features permitted by their role. Sensitive data stored in MongoDB is protected through controlled access and secured API communication.

• **Operational Efficiency and Automation:**
Routine institutional processes such as attendance tracking, task assignment, scheduling, and record maintenance are automated. This reduces administrative workload, eliminates paper-based tasks, and ensures quicker execution of daily operations.

• **Scalability and Modular Architecture:**
Built using the MERN stack, the system offers high scalability. New modules—such as Examination Management, Transport Monitoring, Library Management, or Hostel Allocation—can easily be integrated without affecting existing services. The modular API structure ensures flexible expansion as institutional needs evolve.

- **Customization and Configurability:**

The portal can be customized as per institutional requirements, whether academic rules, workflow structure, or department hierarchy. Admins can modify departmental data, user roles, schedules, and interface settings without altering the system's core logic.

User Roles

- **Administrator(Admin):**

The Administrator oversees the entire ERP Portal and is responsible for system configuration and management. The Admin creates user accounts, assigns roles (Principal, Coordinator, Teacher, Student), manages departments, and controls access rights. This role has complete authority over the platform, ensuring that system settings, module updates, and database maintenance are handled efficiently.

- **Principal:**

The Principal functions as the decision-maker within the portal. This role has access to dashboards displaying institution-wide insights including attendance summaries, teacher performance, and academic progress. The Principal reviews and approves major requests, such as staff leave, task escalations, and departmental reports, ensuring smooth institutional operations.

- **Coordinator:**

The Coordinator acts as the link between Teachers and the Principal. This role handles task scheduling and allocation, verifies attendance and academic data submitted by teachers, and ensures deadlines are met. The Coordinator reviews requests generated by Teachers and forwards them to the Principal for approval when necessary.

- **Teacher:**

The Teacher manages classroom-related activities such as marking attendance, uploading assignments, tracking student progress, and handling academic records. Teachers can communicate with students, submit leave requests, and update task status assigned by the Coordinator.

Technology Stack

The ERP Portal is developed using the **MERN stack**, a modern JavaScript-based framework that supports full-stack web application development.

Frontend:

- **React.js** – Builds a dynamic and responsive user interface.
- **HTML5, CSS3, JavaScript (ES6+)** – Structure, styling, and user interactions.

Backend:

- **Node.js** – Server-side runtime environment for executing backend logic.
- **Express.js** – Framework used to create APIs and manage server routing.

Database:

- **MongoDB** – NoSQL database used to store user data, attendance, tasks, and records.
- **Mongoose** – ODM library for schema definition and database interaction.

Security & Authentication:

- **JWT (JSON Web Token)** – Role-based authentication and secure login.
- **BCrypt.js** – Password hashing and user data protection.

1.2 Objectives of the ERP Portal

The primary objective of the ERP Portal developed using the **MERN Stack (MongoDB, Express.js, React.js, Node.js)** is to digitalize and centralize academic and administrative operations of an institution. The system enhances coordination between different stakeholders—Admin, Principal, Coordinators, and Teachers—ensuring efficient workflow, transparency, and better decision-making through real-time data access.

1. Centralized Data Management

- Maintain a unified MongoDB database that stores all essential institution data such as student details, faculty records, attendance logs, assigned tasks, and academic documents.
- Eliminate duplicate data entry across different departments, ensuring data consistency and reliability.

2. Automation of Institutional Processes

- Automate routine tasks such as attendance marking, timetable handling, task allocation, and report generation.

- Reduce manual dependency and minimize human errors through digitized workflows.

3. Role-Based Access and Authorization

- Implement secure JWT-based authentication, ensuring that each user (Admin, Principal, Coordinator, Teacher) accesses only the functionalities relevant to their role.
- Improve data privacy, avoid unauthorized access, and maintain accountability.

4. Better Communication and Coordination

- Facilitate smooth coordination between Principal, Coordinators, and Teachers through integrated task and workflow modules.
- Reduce delays caused by manual follow-ups, emails, or paperwork.

5. Real-Time Insights and Reporting

- Generate dashboards and visual reports displaying key institutional insights such as attendance rates, pending tasks, and faculty activity.
- Support management in decision-making with accurate, real-time data.

6. Operational Cost Reduction

- Reduce the need for physical paperwork and repeated manual processes.
- Prevent resource misuse and improve overall productivity through streamlined workflows.

7. Scalability and Modular Expansion

- Design the system so new modules (e.g., Exam Management, Transport, Library) can be added without affecting existing functionalities.
- Support future growth of the institution and increasing number of users.

8. Enhanced User Experience

- Provide an intuitive, clean, and easy-to-use React-based UI that reduces training needs and encourages user adoption.
- Improve accessibility and minimize user confusion through consistent and responsive design.

9. Data Security and Integrity

- Preserve data confidentiality through encrypted password storage (bcrypt) and secure session handling.
- Ensure data integrity with controlled update flows and secure API access via Express.js and Node.js.

10. Remote Accessibility

- Enable seamless usage from any device (desktop or mobile) via a browser interface.

- Support hybrid and remote-working scenarios with cloud-hosted database access (MongoDB Atlas).

1.3.1 Member Benefits

1. **Self-Service Portal Access**
   Users such as Teachers and Coordinators can manage their own activities without administrative dependency. They can mark attendance, update academic records, view assigned tasks, raise requests, and track approvals directly through the portal—without needing assistance from the Admin or Principal.
2. **Efficient Task and Workflow Management**
   The portal provides a centralized dashboard where users can view assigned tasks, deadlines, and progress status. Coordinators and Principals can assign and monitor tasks to ensure work is completed on time, while Teachers can update task status, improving overall productivity and accountability.
3. **Integrated Communication and Notifications**
   All important announcements, departmental updates, meeting schedules, and document uploads are available digitally within the portal. Users receive real-time alerts and notifications, eliminating the dependency on manual communication like emails, phone calls, or paperwork.
4. **Remote and Device-Independent Accessibility**
   Since the ERP is web-based and developed using the MERN stack, users can log in securely from any device—laptop, tablet, or mobile—at any time. This flexibility ensures continuous accessibility, ideal for remote work or hybrid learning environments.

1.3.2 Administrator Benefits

1. **Centralized Control and System Management**
   The Admin has complete authority over the ERP Portal and can manage users, departments, permissions, and module configurations from a unified dashboard. This centralized control ensures smooth operations and prevents system-level conflicts or duplication of data.
2. **Role-Based Access and User Privilege Management**
   Using secure authentication (JWT) and role-based access, the Admin can assign

different levels of access to users such as Principal, Coordinators, and Teachers. Each user only views and interacts with the features relevant to their responsibilities, ensuring better security and controlled data access.

3. **Real-Time Monitoring and Analytics Reporting**

   The Admin can generate and view detailed reports related to faculty attendance, student progress, task completion, departmental workflow status, and more. Real-time charts and dashboards provide actionable insights, enabling data-driven planning and timely decision-making.

4. **Automation of Administrative Processes**

   The Admin benefits from automation in processes like user onboarding, notification broadcasting, task allocation, and report generation. This minimizes manual workload, reduces the chance of human errors, and increases the overall operational efficiency of the institution.

1.4 Problem Statement

Many educational institutions and organizations still rely on traditional and partially digitalized methods for managing internal operations. Processes such as attendance tracking, task allocation, academic monitoring, report generation, and communication are often handled manually or through isolated software applications. Due to this lack of integration, institutions face major challenges in terms of efficiency, data accuracy, and workflow transparency.

A system that does not centralize data or automate processes results in operational bottlenecks, limited collaboration, and difficulty in accessing real-time insights necessary for institutional decision-making.

Institutions typically encounter the following issues:

1. Manual and Time-Consuming Administrative Processes

Departments operate independently without a unified system. Attendance records are maintained manually by Teachers, task coordination is handled separately, and approvals from the Principal are often managed through paperwork or emails.

- Leaves are requested using emails or physical forms.
- Attendance summary and reports are compiled manually at the end of each term.

- Task tracking for faculty lacks proper monitoring tools, leading to delays.

Because of manual intervention, the chances of errors increase, and tracking progress becomes difficult and time-consuming.

2. Data Silos and Lack of Integration Across Departments

Each user role maintains its own independent data—Teachers manage attendance and academic updates, Coordinators manage task assignments, and Principals handle approvals. Without a central database:

- Information becomes scattered across different files, emails, and spreadsheets.
- Records are duplicated or outdated due to lack of synchronization.
- Administrators struggle to retrieve data during audits or reporting.

The absence of integration prevents smooth collaboration between roles and leads to inconsistencies in decision-making.

3. Lack of Real-Time Visibility and Tracking

Since most systems are not connected to a real-time dashboard:

- The Admin cannot track how many users have completed their tasks.
- The Principal cannot monitor teacher activity, attendance trends, or academic progress without manually collecting data.
- Teachers and Coordinators are unable to track the status of requests or pending approvals.

This delay in information flow directly impacts institutional planning, evaluation, and decision-making.

4. Ineffective and Unstructured Communication Channels

Communication is fragmented due to reliance on:

- Emails
- Spreadsheets
- Messaging applications
- Verbal updates

There is no workflow transparency. For example:

- Teachers send attendance data via email to Coordinators, who then compile and forward it to the Principal for approval.
- Follow-ups are difficult because no module records the history of task assignments or approvals.

This leads to confusion, delays, and miscommunication among users.

5. Absence of Automated Reporting and Data Analytics

Institutions lack a centralized system to generate:

- Attendance reports
- Task completion reports
- Academic progress summaries

Report generation becomes a manual, repetitive task. Decision-making becomes reactive rather than proactive, and there is no scope for:

- Predictive analysis
- Performance comparison
- Data-driven planning

Without insights, management cannot make informed decisions or track institutional growth effectively.

1.5 Target Audience

The MERN-based ERP Portal is developed to serve multiple types of users across an academic institution. Each user interacts with the portal according to their role and responsibilities, enabling smooth coordination and efficient completion of tasks. The system ensures that every user has access to only the information and functionalities relevant to their role.

1. Teachers (Employees / Faculty Members)

Teachers are the primary users of the ERP system. They access the portal to manage day-to-day academic and administrative activities such as:

- Marking student attendance and uploading academic records.
- Viewing tasks assigned by Coordinators and updating task progress.
- Submitting leave requests and tracking approval status.
- Receiving announcements, notifications, and timetable updates.
- Updating personal information and maintaining their profile.

Teachers benefit from improved task clarity, reduced paperwork, and streamlined approval workflows.

2. Coordinators (Department-Level Managers)

Coordinators act as intermediate managers between Teachers and the Principal. Their responsibilities include:

- Assigning tasks to teachers and monitoring their progress.
- Verifying attendance reports, academic data, and workflow updates.
- Reviewing leave requests before forwarding them to the Principal for decision.
- Ensuring timely completion of academic and administrative activities within their department.

The portal enables Coordinators to have real-time control and visibility over departmental operations.

3. Principal (Decision Authority / Institutional Head)

The Principal uses the ERP Portal primarily for monitoring and decision-making:

- Reviewing and approving requests submitted by Teachers and Coordinators.
- Monitoring academic progress, attendance summaries, and institutional workflow updates.
- Accessing comprehensive reports and dashboards for performance evaluation.
- Making informed decisions based on real-time analytics and insights.

The system allows the Principal to oversee the institution efficiently with a data-driven approach.

4. Administrator (System Controller / Super Admin)

The Admin manages the entire ERP portal and ensures continuous system functionality:

- Creating user accounts and assigning role-based access permissions.
- Managing modules, system settings, and portal configuration.
- Monitoring overall system activity and ensuring data security.
- Generating institutional reports and maintaining system integrity.

The Admin ensures smooth functioning of the ERP system at both technical and operational levels.

5. IT Support Team (Technical Maintenance Team)

IT support handles technical workflows associated with the portal:

- Monitoring application performance and resolving system issues.
- Managing backups, updates, and server/database maintenance.
- Ensuring system security, uptime, and technical troubleshooting.

They ensure that the ERP Portal remains stable, secure, and available at all times.

1.6 Project Significance

The MERN-based ERP Portal plays a significant role in transforming the operational and academic processes of an educational institution. By integrating key functions—such as task management, attendance monitoring, role-based workflows, and reporting—into a centralized web platform, the system enhances institutional transparency, reduces administrative load, and promotes performance-driven decision-making.

This ERP solution brings together Admin, Principal, Coordinator, and Teacher modules into a unified system that streamlines collaboration, reduces redundancies, and ensures real-time data availability. Its implementation eliminates fragmented processes and supports long-term institutional growth through automation, scalability, and secure data handling.

1. Improved Efficiency and Productivity

Routine academic and administrative activities—such as attendance entry, task assignment, leave approval, and workflow tracking—are automated within the ERP.

- Teachers can record attendance and update progress seamlessly.
- Coordinators can assign and track tasks easily.
- Principals can approve requests without delays.

Automation reduces the dependency on manual paperwork, minimizes errors, and increases productivity across departments.

2. Centralized and Real-Time Data Accessibility

All institutional data—attendance records, task status, approvals, announcements—is stored in **MongoDB**, establishing a single source of truth.

- Users can access updated information at any time from any device.
- Eliminates redundant data storage across departments.

Centralization strengthens collaboration and ensures consistency in decision-making.

3. Enhanced Decision-Making Through Analytics

The portal provides dashboards and reports for transparent insights:

- Principals can analyze faculty performance and task progress.
- Admins can monitor system-wide usage and workflows.
- Coordinators can track departmental performance.

Real-time analytics allow management to make informed, data-backed decisions.

4. Reduced Operational and Administrative Costs

By digitizing documentation and workflow management:

- Paper usage decreases.
- Manual file handling and redundant data entry are eliminated.
- Resource utilization becomes optimal.

This leads to long-term financial savings for the institution.

5. Scalability and Flexibility

Built on the **MERN stack**, the system supports modular expansions:

- Additional features (Exam Management, Student Portal, Transport Management, etc.) can be introduced without affecting the existing structure.
- The platform adjusts easily as the organization grows.

The system remains future-ready and adaptable.

6. Data Security and Compliance

The ERP ensures secure handling of sensitive institutional data:

- Passwords are encrypted using bcrypt.
- Authentication and user access are controlled through JWT-based role management.
- Regular database backups ensure data availability and protection.

This ensures confidentiality, integrity, and compliance with institutional and regulatory data handling policies.

7. Enhanced User Satisfaction and Engagement

Role-based dashboards and self-service accessibility empower users:

- Teachers manage tasks independently.
- Coordinators track workflow without manual follow-ups.
- Principals access reports instantly without requesting data from departments.

Transparent processes improve user engagement and satisfaction.

8. Real-Time Communication and Collaboration

Notifications, reminders, and updates are shared instantly across the system:

- Coordinators can broadcast departmental announcements.

- Teachers receive immediate task and timetable updates.
- Principals can share institutional notices.

This reduces communication delays and strengthens teamwork.

1.7 Limitations of the System

Although the MERN-based ERP Portal significantly improves institutional efficiency by automating workflows and centralizing data, the system has certain limitations that could affect implementation, performance, or user experience. These limitations arise due to technological constraints, organizational challenges, and adoption factors.

1. High Initial Implementation and Deployment Cost

Implementing an ERP system requires investment in:

- Hosting and server resources (e.g., MongoDB Atlas, cloud deployment).
- User onboarding and training programs.
- Initial development and customization efforts.

For smaller institutions with limited budgets, these initial costs may seem prohibitive, even though the long-term benefits eventually outweigh the expenditure.

2. Complexity in Configuration and Customization

Although the system is designed with modular architecture, initial setup may require:

- Role configuration (Admin, Principal, Coordinator, Teacher)
- Defining workflows, rules, and approval processes
- Integration with institution-specific processes

Customization requires technical expertise in the MERN stack (MongoDB, Express.js, React.js, Node.js). Without proper configuration, the ERP may not align perfectly with operational needs.

3. Training and User Adaptation Challenges

Users previously accustomed to manual record keeping or basic tools (spreadsheets, email communication) may face:

- Difficulty in adapting to digital workflows
- A learning curve in understanding system navigation
- Temporary reduction in productivity during transition

Successful adoption requires training, awareness, and change-management strategies.

4. Risk of System Downtime and Technical Glitches

As a web-based system, temporary downtime may occur due to:

- Server issues or outages
- Backend or database failures
- Bug fixes during updates

Such interruptions may affect daily activities like attendance tracking, task updates, or approvals, particularly when the ERP is the primary workflow platform.

5. Limited Third-Party Integration

The ERP is optimized for internal data flow but may face difficulty integrating with:

- Legacy systems (existing attendance machines, academic record systems)
- External software (finance/accounting tools, biometric machines)

Integration requires APIs or custom development, which may increase cost and effort.

6. Scalability Challenges with Increasing Data Load

Although the MERN stack supports scalability, large organizations with:

- Very high user counts
- Large data storage volume (attendance history, documents)
- Concurrent heavy usage

may experience performance slowdowns. Scaling the system may require upgrading hosting plans, optimizing databases, or implementing distributed architecture.

7. User Resistance to Change

Resistance may occur when:

- Users are comfortable with existing manual or semi-digital methods

- Automation disrupts existing workflow habits

Lack of acceptance may lead to underutilization of system features, reducing expected benefits.

8. Dependence on Internet Connectivity

Since the ERP is web-based, uninterrupted internet access is required. In environments with:

- Poor internet connectivity
- Limited network infrastructure

users may experience accessibility issues, affecting productivity and real-time operations (attendance, announcements, task status, etc.).

9. Security Risks and Data Exposure

Despite the use of JWT authentication, bcrypt password encryption, and secure role-based access:

- Increased online data visibility introduces cyber risks
- Potential vulnerabilities may arise due to external threats or misconfigurations

Continuous monitoring, backups, firewalls, and audits are required to maintain data security.

2. FEASIBILITY STUDY / LITERATURE REVIEW

2.1 Technical Feasibility

The technical feasibility of implementing the ERP Portal for Organizational Resource Management evaluates whether the required technology, software, hardware, network infrastructure, and technical skills are sufficient to develop, deploy, and maintain the system successfully. This feasibility study ensures that the proposed solution can be practically implemented using available technological resources and can scale with future organizational needs.

1. Technology Stack

The ERP portal will be developed using the **MERN stack**, a modern, scalable, and open-source technology stack that supports end-to-end development:

- **Frontend – React.js**
    - Provides a responsive, fast, and interactive user interface.
    - Component-based architecture allows reuse of UI components (dashboard, forms, cards, tables, etc.).
    - Supports state management for real-time updates of attendance, payroll data, and tasks.
- **Backend – Node.js with Express.js**
    - Handles business logic, CRUD operations, authentication, and communication with the database.
    - Supports large volumes of concurrent users, making it ideal for enterprise-level portals.
- **Database – MongoDB (NoSQL)**
    - Stores employee details, attendance records, payroll, and inventory data securely.
    - Flexible schema allows addition of new modules (HR, Finance, Inventory, Tasks) without restructuring existing data.
- **Authentication & Security Technologies**
    - **JWT (JSON Web Tokens)** for secure role-based access control.
    - **BCrypt** for password hashing and secure credential storage.
    - HTTPS + CORS to ensure secure communication between client and server.

Benefit: MERN is cost-effective, open-source, scalable, and suitable for cloud deployment.

2. Infrastructure Requirements

The ERP portal will be hosted on cloud servers to ensure continuous availability, data security, and performance.

- **Cloud Hosting (Preferred):**
  - Platforms such as AWS, Render, Azure, or Railway provide automatic backup, auto-scaling, and load balancing.
  - No hardware investment required, reducing initial expenses.
- **On-Premise Hosting (Optional):**
  - Requires dedicated servers, storage, network switches, UPS backup, and security firewalls.
  - Recommended for organizations with strict data compliance requirements.

User Devices:

- Accessible on desktop, laptop, tablet, and mobile devices.
- Works using web browsers like Chrome, Edge, Firefox—no installation required.

Benefit: Users can access the ERP portal from anywhere with an internet connection.

3. Integration with Existing Systems

The ERP portal is designed to support seamless integration with:

- Biometric attendance tracking devices
- Payroll systems
- Email notification systems (SMTP, Gmail API)
- Cloud storage for document uploads (AWS S3, Firebase Storage)

Using **REST APIs**, data can be imported/exported between the ERP portal and existing software used by the organization.
Data migration tools will be used to import old employee records, payroll data, and leave statistics into MongoDB.

Benefit: Smooth transition without interrupting current business operations.

4. Scalability and Flexibility

The ERP system is modular, meaning new features or modules can be added without modifying the existing codebase.

- Horizontal scalability ensures that as the number of users increases, more servers can be added.
- MongoDB supports distributed storage and can handle growing data efficiently.
- The system can extend to support additional modules in the future, such as:
    - Transport Management
    - Hostel/Facility Management
    - Examination/Academic Management

Benefit: The ERP portal grows with the organization.

5. Development Team Expertise

To build and maintain the ERP portal, the development team must possess expertise in:

- MERN stack development (MongoDB, Express.js, React.js, Node.js)
- Cloud deployment & DevOps (AWS, Docker, CI/CD pipelines)
- Database administration and performance optimization
- Cybersecurity practices (role-based access control, token-based authentication)
- REST API integration and versioning

Additionally, the team must be capable of maintaining documentation, providing user support, and implementing software updates.

Benefit: Skilled developers ensure reliable and high-performance system implementation.

2.2 Economic Feasibility

Economic feasibility evaluates whether developing and implementing the ERP Portal for Organizational Resource Management is financially viable for the organization. It examines both the initial investment and recurring maintenance costs and compares them with the long-term benefits and cost savings the system will generate. This analysis helps determine if the ERP project is a cost-effective and sustainable solution for the organization.

1. Cost Analysis

The cost of the ERP portal implementation can be categorized into **initial development cost** and **ongoing operational cost**.

Initial Development Costs include:

- **Software Development and Customization**
  Cost of designing and developing features such as user authentication, dashboard, attendance tracking, payroll, inventory, and reporting modules.
  Since the project uses the MERN stack (MongoDB, Express.js, React.js, Node.js), which is open-source, there are no software licensing costs, reducing the initial expenditure.

- **Hardware or Hosting Infrastructure**
  If hosted on cloud platforms (AWS, Render, Railway, Azure), the cost involves server instances and database storage.
  If hosted on-premise, the organization needs to purchase servers, networking equipment, and backup devices.

- **Technical Staff Hiring or Training Cost**
  Includes hiring MERN developers or training existing staff to manage and use the system.

- **Data Migration and Integration**
  Cost associated with migrating existing data (manual files or other software data) into the new ERP database.

- **User Training and Documentation**
  Training users (admin, coordinator, teacher, principal) to navigate the portal and providing manuals or tutorials.

Ongoing Operational Costs include:

- **Maintenance and System Updates**
  Enhancements, bug fixes, module upgrades, and performance optimization to keep the system stable and secure.

- **Technical Support and Monitoring**
  Continuous monitoring of uptime, response time, and database performance.

- **Server Hosting / Cloud Subscription Charges**
  Cost of cloud hosting, storage expansion, and backup services if using cloud deployment.

- **Renewal of Third-Party Services (if used)**
  Fees for email API services, SMS gateways, or code repositories (optional).

2. Benefit Analysis

Although the ERP portal may involve an initial investment, it delivers significant financial and operational returns, such as:

- **Reduction in Manual Workload**
  Automation of attendance, payroll generation, task assignments, and leave approval reduces dependency on manual paperwork and clerical staff.

- **Increased Productivity**
  Employees can focus on productive tasks instead of administrative work. Coordinators and principals get real-time data without delays.

- **Cost Savings on Stationery and Physical Records**
  Digital workflows eliminate printing, storing, and managing paper documents.

- **Faster and Smarter Decision-Making**
  Real-time dashboards and reports allow management to take timely and data-driven decisions, improving operational control.

- **Scalability**
  As the organization grows, the same ERP system can accommodate new modules and users without proportional cost increases.

3. Return on Investment (ROI)

The ERP system promises a high ROI because the benefits increase continuously while the costs remain stable.

- Reduced administrative burden leads to lower operational costs.
- Digital automation improves efficiency and reduces delays.
- Employee satisfaction improves due to transparency and self-service features.
- Long-term cost savings outweigh the development and maintenance investment.

Even though the initial implementation cost may appear significant, the long-term reduction in recurring costs makes the ERP portal financially viable and beneficial.

4. Risk Assessment and Mitigation

- **Risk: High initial cost investment**
  Mitigation: Use open-source technologies like MongoDB, React, Node.js, and deploy in modules to spread development cost over time.

- **Risk: Features not used effectively by users**
  Mitigation: Conduct regular training sessions for principals, teachers, and coordinators, and implement the system in phases.

- **Risk: Budget overrun during development**
  Mitigation: Create a detailed project plan, define scope clearly, prioritize modules, and control unnecessary customizations.

2.3 Market Research

Market research is essential to justify the need for developing the ERP Portal for Organizational Resource Management. This section analyzes current market demand, existing ERP competitors, user expectations, and the competitive advantage of the MERN-based ERP system. The research validates that there is increasing demand for a customizable, cost-effective, and user-friendly ERP solution, especially in educational and institutional environments.

1. Growing Demand for ERP Solutions

Over the last few years, organizations across various domains—such as education, corporate offices, healthcare, manufacturing, logistics, and IT services—have been actively shifting from manual processes to automated cloud-based ERP solutions. Digital transformation has become essential for improving productivity, reducing operational costs, and maintaining accurate records.

Market reports confirm this trend:

- According to **Statista** and **Grand View Research**, the global ERP market is projected to grow at a **CAGR of more than 10% by 2030**.
- Adoption is rising rapidly among **small and medium-sized enterprises (SMEs)** due to:
    - Low maintenance cost
    - Cloud-based deployment
    - Minimal need for IT infrastructure

The increasing demand highlights an opportunity for ERP systems that offer:

- Centralization of workflows
- Automation of tasks
- Ease of access for multiple user roles

This validates the relevance and potential success of the newly developed ERP portal.

2. Existing Solutions in the Market

Several ERP solutions exist in the market, including:

- **SAP ERP** – Comprehensive but expensive, complex, and suitable mainly for large enterprises.
- **Oracle NetSuite** – Cloud-based and feature-rich but cost-intensive for smaller organizations.
- **Zoho ERP / Odoo ERP** – More affordable and modular, but still require licensing fees and technical knowledge for customization.

Although these systems are powerful, they exhibit common drawbacks:

- High subscription and licensing cost
- Complexity, making training and adoption difficult
- Limited customization for academic or institutional environments
- Overloaded features that may be unnecessary for smaller organizations

This highlights a gap in the market for an ERP solution that is:

- Flexible and customizable
- Cost-effective
- Tailored specifically for institutional resource planning

3. Target Users' Needs and Expectations

To understand real user needs, informal surveys, online community feedback, and user interviews were considered. Across all feedback sources, the requirements were consistent:

Users expect an ERP system that:

- Provides **a unified portal** for managing tasks like attendance, payroll, leave requests, academic records, asset tracking, and communication.
- Offers **role-based access**, ensuring users (Admin, Principal, Coordinator, Teacher) view only relevant modules.
- Provides **real-time data access** without delays.
- Works on both desktop and mobile devices with a responsive user interface.
- Supports **integration** with tools like email, biometric devices, or academic learning platforms.
- Allows **customizable dashboards** for easy monitoring of data.

These insights influenced the design of the MERN-based ERP portal, focusing on simplicity, modularity, and user accessibility.

4. Competitive Advantage of the Proposed ERP Portal

The MERN-based ERP system stands out because it is designed specifically for institutional workflow automation—unlike generic ERP systems that try to fit multiple industries.

The proposed ERP Portal offers:

- **Customization**

  Modules like Attendance Management, Task Assignment, Payroll, and Inventory are fully customizable based on institutional workflow.
- **Cost Efficiency**

  Built using MERN—an open-source tech stack—making it significantly cheaper than licensed systems such as SAP or Oracle.
- **Scalable Modular Architecture**

  New modules (e.g., Exam Management, Transport Management) can be added easily without affecting existing features.
- **User-Centered Design**

  Minimal learning curve, intuitive dashboard, fast navigation, clean UI, and mobile responsiveness.

- **Fast Deployment and Installation**

  Since the MERN stack supports API-based modular development, rollout can be done in phases:

  - Start with attendance and user management
  - Add payroll, inventory, or additional modules later

Thus, the newly developed ERP system fills a substantial market gap for a **customizable, affordable, and user-friendly ERP portal** built using modern web technologies.

3. PROJECT OBJECTIVE

The main objective of the ERP Portal for Organizational Resource Management, developed using the **MERN (MongoDB, Express.js, React.js, Node.js) stack**, is to build a unified and automated platform that centralizes academic and administrative processes. The system is designed to replace traditional paper-based workflows and scattered tools with a single digital portal that connects **Admin, Principal, Coordinator, and Teacher** roles through structured automation, secure access, and real-time data sharing.

This project aims to ensure operational transparency, increase productivity, reduce manual workload, and support data-driven decision-making.

1. Centralized Management System

To integrate multiple academic and administrative functions into one digital platform.

The portal will consolidate:

- User and staff management
- Attendance tracking
- Task scheduling and academic activity monitoring
- Internal notifications and announcements
- Document uploads and resource sharing

Instead of using separate tools such as spreadsheets, emails, WhatsApp groups, and registers, all operations will happen within the ERP portal.

This creates *one source of truth* for all institutional data.

2. Workflow Automation to Improve Efficiency

To automate repetitive and time-consuming tasks, reducing manual dependency.

Examples of automation implemented:

- Teacher logs attendance → Coordinator verifies → Principal reviews → Admin monitors
- Coordinator assigns tasks → Teacher updates progress → Principal gets notified
- Principal approval workflow for reports, academic events, and schedules

Automation benefits include:

- Reduced chances of human error
- Reduced paperwork and manual entries
- Faster execution of institutional operations

3. Role-Based Access Control (RBAC)

To implement secure and structured access permissions depending on user type.

Each role receives a dashboard designed for their needs:

- **Admin:** Full access (create user accounts, manage modules, view reports)
- **Principal:** Dashboard for approvals, insights, and overall monitoring
- **Coordinator:** Manage tasks, monitor department-level teacher activity
- **Teacher:** Mark attendance, update task status, receive announcements

RBAC ensures security by limiting access to data that is relevant to that role and preventing unauthorized actions.

4. Intuitive and User-Friendly Interface

To ensure that users with little technical background can operate the portal easily.

Key interface objectives:

- Modern, responsive React UI
- Mobile-friendly layout
- Clear navigation structure
- Simplified forms and dashboards

The system will eliminate technical barriers and encourage digital adoption across all academic staff.

5. Real-Time Data, Reporting, and Analytics

To support informed decision-making by providing insights through:

- Dashboards with live activity updates
- Visual graphs for attendance and task progress
- Automated report generation

Examples:

- Admin and Principal can track attendance trends
- Coordinator can monitor teacher performance and deadlines
- Teachers can access their schedules and tasks instantly

This eliminates the delay associated with manually compiling reports.

---

6. Scalable and Modular System Architecture

To ensure long-term usability and easy expansion.

The architecture must allow:

- Adding new modules (Exams, Payroll, Transport, Hostel, Library, etc.)
- Connecting to third-party services (email automation, SMS notifications)
- Migration to larger infrastructures as institution size increases

The MERN stack enables horizontal and vertical scalability without rewriting core logic.

7. Secure and Reliable System Operations

To protect sensitive institutional and employee data.

Security features include:

- Encrypted password storage (bcrypt hashing)
- JWT authentication for secure login sessions
- Sensitive data protection using access restrictions in MongoDB
- Regular automated database backups
- Audit logs for tracking user actions

This strengthens data confidentiality and ensures compliance with modern security standards.

8. Customizability to Fit Any Institution

To enable the system to adapt to organizational needs instead of forcing a predefined process.

The ERP portal supports:

- Adding/removing modules based on requirement
- Renaming features based on institution terminology
- Custom workflows (e.g., approval workflows, different reporting formats)

This makes it usable for schools, colleges, training institutes, coaching centers, and private organizations.

4. HARDWARE AND SOFTWARE REQUIREMENTS

4.1 Hardware Requirements

For the **ERP Portal for Organizational Resource Management** project, the following hardware components are recommended to ensure efficient development, testing, and deployment:

- **Processor (CPU):**
  A minimum of an Intel Core i3 or AMD Ryzen 3 processor is sufficient for basic development and testing. However, for optimal performance during multitasking and running local servers, an Intel Core i5 or AMD Ryzen 5 processor (or higher) is recommended.
- **RAM:**
  At least 4 GB of RAM is required for basic development tasks such as coding and compiling. For smoother performance, especially when running databases, backend services, and browsers simultaneously, 8 GB of RAM or more is ideal.
- **Storage:**
  A minimum of 100 GB of storage space is necessary to hold project files, dependencies, databases, and logs. An SSD (Solid State Drive) of 256 GB or more is highly recommended for faster boot times and application performance.
- **Monitor:**
  A standard display with a resolution of 1366×768 pixels is the minimum requirement. However, for better productivity and UI/UX design accuracy, a Full HD (1920×1080) or higher resolution monitor is preferred.
- **Internet Connection:**
  A reliable broadband internet connection is required for downloading dependencies, accessing repositories, API integration, and cloud deployment. For continuous collaboration and cloud testing, a stable high-speed internet connection is strongly recommended.
- **Backup Power Supply (Optional but Recommended):**
  A UPS (Uninterruptible Power Supply) is advised to avoid data loss or service interruption during power outages, especially during development or local server testing.

4.2 Software Requirements

TheThe software requirements define all tools, platforms, and technologies necessary for developing, testing, and deploying the **ERP Portal for Organizational Resource Management**. The project is built using the **MERN Stack (MongoDB, Express.js, React.js, Node.js)**, which provides full-stack JavaScript development, scalability, and flexibility.

System Requirements

The ERP application can be developed and deployed on any of the following operating systems:

- **Windows 10 / Windows 11**
- **Ubuntu Linux (20.04 or above)**
- **macOS (Monterey or above)**

These operating systems support the required development tools, databases, and runtime environments without compatibility issues.

1. Backend Development

- **Node.js (v18 or above)**
    - Provides the runtime environment to execute JavaScript on the server.
- **Express.js (v4 or above)**
    - A minimal and flexible backend framework used to create RESTful APIs, manage authentication, handle routing, and process database operations.

2. Frontend Development

- **React.js (v18 or above)**
    - Used to build an interactive, user-friendly, responsive, and component-based user interface.
- **HTML5**
    - Structures webpage content and dynamic layout containers.
- **CSS3**
    - Provides responsive design, styling, animations, and layout formatting using frameworks like Bootstrap or Tailwind.
- **JavaScript (ES6+)**
    - Enables dynamic behavior, form validation, UI updates, and API communication in the frontend.

3. Database

- **MongoDB (v6 or above)**
    - A NoSQL document-oriented database used to store user data, attendance, tasks, logs, access roles, etc.
    - Supports scalability and fast data access.

Optionally, **MongoDB Compass** or **Atlas** (cloud version) can be used for database visualization and management.

4. API & Testing Tools

- **Postman (Optional)**
  - Helps in testing REST API endpoints, validating data transfer between frontend and backend, and debugging requests/responses.

5. IDEs / Code Editors

Any modern IDE can be used based on developer preference:

- **Visual Studio Code (Recommended)**
- **WebStorm**
- **Sublime Text**

VS Code extensions like ESLint, Prettier, and MongoDB Tools enhance development productivity.

6. Version Control & Collaboration

- **Git (Version Control System)**
  - Tracks project changes and manages local commits.
- **GitHub / GitLab / Bitbucket (Source Code Repository)**
  - Used for remote collaboration, repository hosting, issue tracking, and continuous integration.

7. Deployment Tools

- **Node Package Manager (npm)** or **Yarn** for dependency management
- Cloud platforms (Optional, based on deployment preference):
  - **Vercel / Netlify** → Host frontend (React)
  - **Render / AWS / Azure / Heroku** → Host backend (Node + Express)
  - **MongoDB Atlas** → Cloud database hosting

4.3 Other Tools

Apart from core software and development environments, certain supporting tools are required to enhance development efficiency, ensure collaboration, and streamline deployment. These tools assist in UI/UX design, application packaging, version control, dependency management, and production deployment.

1. Docker

Docker is used to **containerize the ERP application**, ensuring that the backend, frontend, and database run consistently across all environments (development, testing, staging, and production).

Purpose and Benefits:

- Enables developers to package the entire application along with its dependencies into lightweight containers.
- Eliminates environment-based issues such as "it works on my system but not on the server."
- Allows for easy deployment and scaling of microservices or independent modules.

Usage in the ERP Portal:

- Backend (Node.js + Express) container
- Frontend (React build) container
- MongoDB database container

2. Figma / Adobe XD (UI/UX Design Tools)

Figma or Adobe XD will be used for creating **wireframes, UI mockups, and prototypes** of the ERP portal before development.

Purpose and Benefits:

- Helps visualize the layout, screen flow, and user experience before coding.
- Allows quick design changes based on feedback.
- Collaboration-friendly — multiple team members can work and comment in real-time.

Usage in the ERP Portal:

- Designing dashboard layouts (Admin View, Employee View)
- Creating UI components like forms, tables, and charts before coding them
- Building prototypes to demonstrate workflow to stakeholders

3. Node.js & npm (Node Package Manager)

Node.js and npm are essential tools for managing packages and running both frontend and backend environments.

Purpose and Benefits:

- Node.js enables server-side JavaScript execution for backend development (Express.js).
- npm manages third-party libraries and dependencies required by the project.

Usage in the ERP Portal:

- Installing backend packages (Express.js, JWT authentication, bcrypt, mongoose, etc.)
- Running the frontend React application and installing UI libraries (Material UI, Bootstrap)
- Managing build processes and automation scripts

5. PROJECT FLOW

5.1 Development Methodology

The ERP Portal for Organizational Resource Management is developed using an **Agile Software Development Methodology**, specifically the **Iterative Agile Sprint Model**. This model ensures that the project is delivered in manageable phases, allows continuous feedback, and adapts to evolving requirements from stakeholders.

1. Requirement Gathering and Analysis

This is the foundation of the project. The objective is to fully understand what the organization expects from the ERP portal.

Activities involved:

- Conduct meetings and interviews with stakeholders such as HR, finance department heads, administrators, and IT staff.
- Identify pain points with current processes, such as manual attendance tracking, scattered data, or lack of reporting tools.
- Identify mandatory system modules such as:
  - User Role Management (Admin, Employee, HR)
  - Attendance & Leave Module
  - Payroll & Salary Module
  - Asset / Inventory Management Module
  - Report Generation

Outputs of this phase:

- *Software Requirement Specification (SRS)* — contains functional and non-functional requirements.
- *User Persona Definitions* — understanding different types of users and their goals.
- *Feature Prioritization* — decides what features must be developed first (e.g., User Authentication → Attendance → Payroll).

2. System Design and Architecture

This phase converts the business requirements into a technical blueprint.

Activities involved:

- Designing the system architecture (3-tier architecture):
  - **Frontend Layer** → React.js (UI Interface)
  - **Backend Layer** → Spring Boot (API + Business Logic)
  - **Database Layer** → MySQL (Data Storage)

- Designing ER diagrams, database schema, and relationships (users, roles, attendance, payroll).
- Designing UI mockups using *Figma / Adobe XD*.
- Identifying security and integration requirements (JWT-based authentication, role-based access control).

Outputs of this phase:

- *Architecture Diagrams (HLD + LLD)*
- *UI Wireframes / Prototypes*
- *Database schema and ER Diagram*

3. Development Phase (Agile Sprint Cycle)

Development is executed in **short sprints of 2–3 weeks**. Each sprint focuses on delivering one or more working modules.

Activities involved:

- Setting sprint goals using sprint planning sessions.
- Developers work on:
  - Backend APIs for business logic (Spring Boot)
  - Database schema creation (MySQL)
  - Responsive frontend screens (React.js)
- Version control is handled using **Git and GitHub** (branching model: feature → develop → main).

Sprint Deliverables:

- Sprint 1 → Login, Registration, Role-based Dashboard
- Sprint 2 → Attendance & Leave Module
- Sprint 3 → Payroll + Asset Management
- Sprint 4 → Reporting Module + Enhancements

Regular **sprint reviews** are held to demonstrate progress and collect feedback for improvement.

4. Testing Phase

Testing is continuous and parallel to development to ensure quality.

Types of Testing Performed:

- **Unit Testing (Backend APIs)** – ensures individual functions/modules work properly.
- **Integration Testing** – validates the connection between frontend, backend, and database.
- **Functional Testing** – verifies that the system performs expected business operations.

- **UI/UX Testing** – checks responsiveness, user flow, and design consistency.
- **User Acceptance Testing (UAT)** – real users test the system before deployment.

Tools used: *Postman (API testing), Jest/React Testing Library (frontend testing)*

Output of this phase:

- Bug reports
- Test case documentation
- Final approval for production release

5. Deployment Phase

Once the application passes UAT, it is prepared for live deployment.

Activities involved:

- Setting up infrastructure on cloud or internal server (Docker used for deployment consistency)
- Configuring **CI/CD pipeline** using GitHub Actions or Jenkins
- Setting up SSL certificate (HTTPS), domain routing, and database backup policies

Deployment Environments:

- **Staging / Testing Server** → For internal review
- **Production Server** → Live environment for end users

6. Maintenance and Updates

After deployment, the system enters a long-term maintenance cycle.

Activities involved:

- Fixing user-reported bugs
- Updating dependencies and applying security patches
- Optimizing performance and adding new modules (new workflow automation, more reports, etc.)
- Monitoring API performance and server uptime

The system evolves with organizational needs, ensuring long-term sustainability.

Data Flow Diagram Level 0 and 1

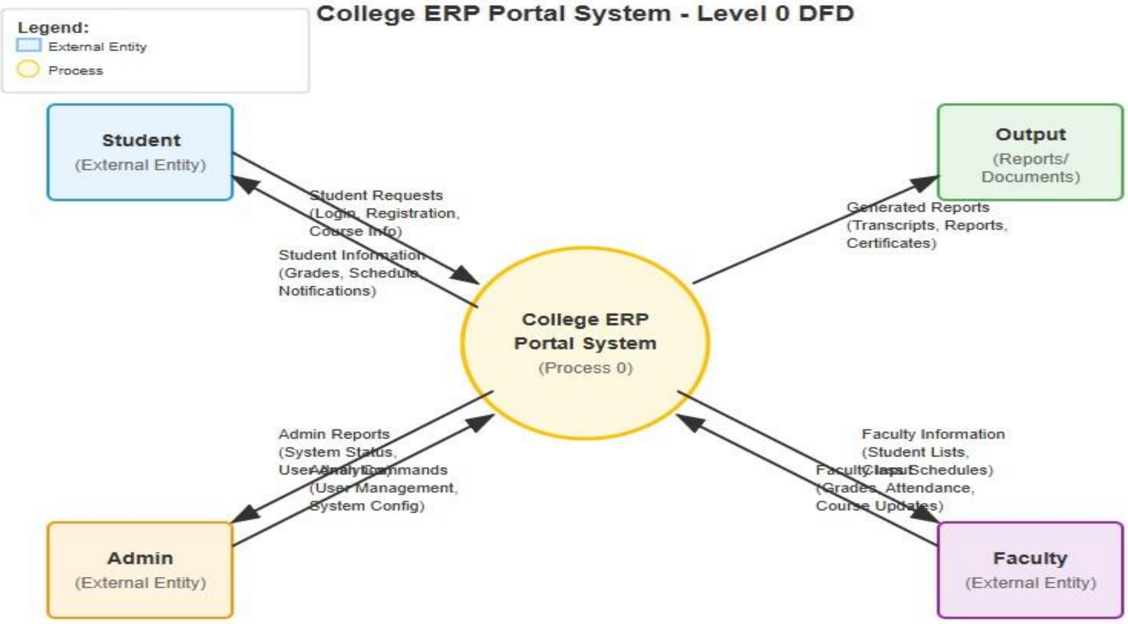


Fig. 1. <u>DFD LEVEL 0</u>

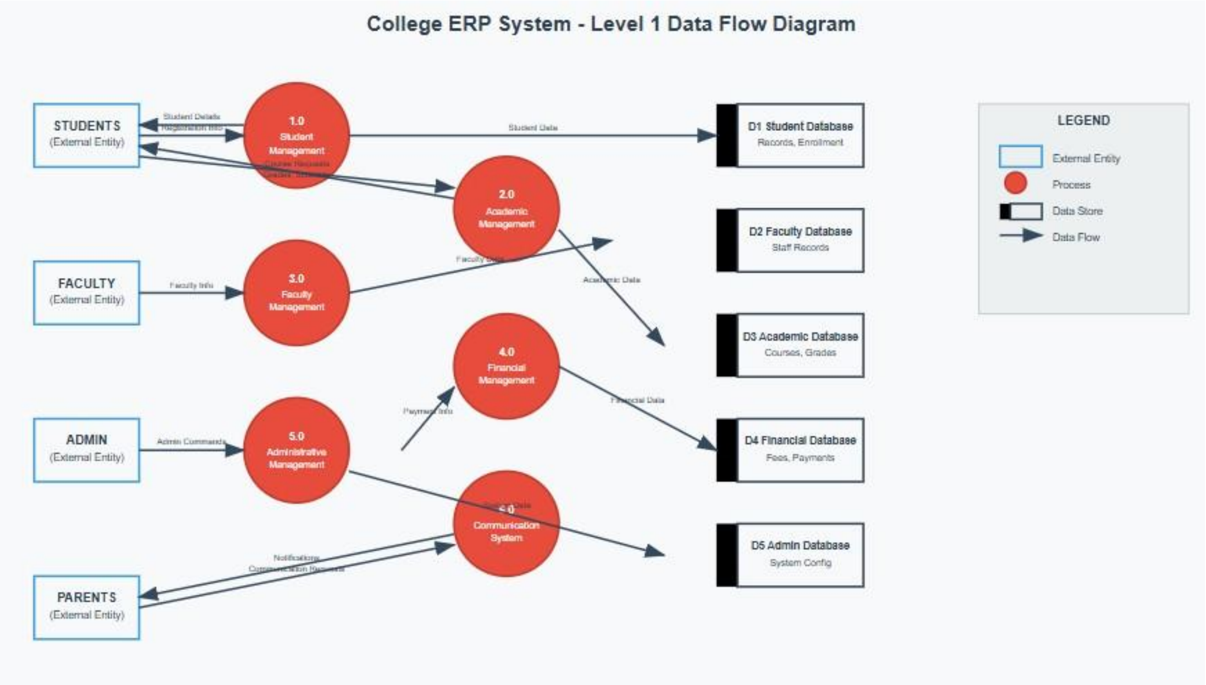


Fig.2. DFD LEVEL 1

6. Summary of ERP Portal

The ERP (Enterprise Resource Planning) Portal for Organizational Resource Management is a comprehensive web-based solution developed using the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. The system integrates various organizational operations into a single platform, eliminating manual processes, improving efficiency, and enabling smooth inter-department communication.

Purpose of the System

The primary goal of the ERP Portal is to streamline daily administrative tasks and provide a centralized digital system for managing employees, attendance, payroll, inventory, and departmental tasks. By automating workflow processes, the system ensures that data is accurate, accessible, and updated in real time.

User Experience

The system provides a modern and user-friendly interface. Employees can log in and perform functions through a dedicated dashboard. They can:

- Mark attendance online and view their attendance history.
- Apply for leave and track approval status.
- Check assigned tasks, update progress, and view deadlines.
- Access salary records and download payslips.
- Manage and update personal information.

This empowers employees with self-service capability and reduces dependency on HR staff.

Administrative and Manager Functionality

Administrators and department heads have advanced privileges. They can:

- Manage user accounts, roles, and module access.
- Approve leave requests and monitor attendance patterns.
- Assign tasks to employees and track progress of work.
- Generate payroll based on attendance and leave deductions.
- Monitor resource utilization and manage inventory.
- View and download detailed analytics and reports to assist decision-making.

This improves accountability and enables informed decision-making through real-time insights.

Technical Overview (MERN Stack)

The frontend is developed using React.js, which ensures smooth navigation, responsiveness, and dynamic interaction. Node.js and Express.js handle backend logic, API routing, authentication, and business workflows. MongoDB is used as the database to store all system-related records, allowing fast, flexible, and scalable storage of employee data, attendance logs, payroll data, tasks, and inventory information.

Authentication and security features are implemented using JSON Web Tokens (JWT), ensuring secure login and proper role-based access control. The system architecture efficiently connects the frontend, backend, and database, ensuring real-time data flow across all modules.

Key Modules of the ERP Portal

- Attendance Management: Captures daily attendance and logs timestamps securely.
- Leave Management: Enables employees to request leaves and managers to approve or reject them.
- Task and Project Tracking: Allows assignment of tasks with status tracking and completion analytics.
- Payroll Automation: Automatically calculates salary based on attendance records, deductions, and leave balance.
- Performance Monitoring: Tracks productivity and supports data-driven evaluations.

Benefits

The ERP Portal minimizes the need for manual paperwork and eliminates fragmented departmental systems. It ensures transparency, enhances productivity, and improves communication within the organization. The portal is scalable and can easily be extended by adding new modules according to future organizational needs.

7. PROJECT OUTCOME

7.1 System Features

The MERN-based ERP Portal is designed to digitalize and streamline internal operations within an organization, specifically focusing on academic or institutional environments. The system delivers a simplified and automated platform where all departments can collaborate, share real-time information, and perform their operations efficiently. Developed using MongoDB, Express.js, React.js, and Node.js, the ERP Portal ensures high performance, scalability, and secure data handling.

Below are the major system features achieved through this project:

Secure Authentication & Role-Based Access

- The system provides a secure login process using JWT authentication.
- Access and screen visibility are controlled based on roles: Admin, Principal, Teacher, and Coordinator.
- Users can only access modules and data permitted to their role, ensuring security and privacy.

User & Employee Management

- Maintains a centralized database of all staff members.
- Admin/Principal can add, update, or disable user accounts.
- Teachers and Coordinators can update their profiles and manage assigned tasks.

Attendance & Leave Management

- Attendance tracking for staff with real-time updates.
- Staff can apply for leave directly on the portal.
- Principal/Admin can approve or reject leave requests.
- Automatic calculation of attendance percentage and leave balance.

Task / Work Allocation & Progress Tracking

- Principal and Coordinator can assign tasks to teachers.
- Teachers can update task status, attach files, and add progress notes.
- Coordinators can monitor task performance and upcoming deadlines.
- Supports transparency and accountability in task execution.

Payroll Management (Automated)

- Payroll is auto-generated based on attendance and approved leaves.
- Calculates deductions, bonuses (if applicable), and final salary.
- Generates downloadable salary slips via the portal.

Inventory / Resource Tracking

- Tracks institutional assets such as equipment, lab resources, or departmental materials.
- Allows departments to raise purchase or resource issuance requests.
- Maintains logs of resource utilization to avoid misplacement or wastage.

Reports, Insights, and Analytics

- Admin/Principal can generate real-time reports on:
  - Attendance trends
  - Task completion progress
  - Payroll summary
  - Resource usage
- Visual dashboards using charts and graphs built through React.js libraries.

Notification and Alert System

- Sends portal notifications for:
  - Task assigned
  - Leave approval/rejection
  - Payroll/payslip generated
  - Important announcements
- Ensures transparency and informed communication across departments.

Secure Document Storage

- Supports uploading and storing documents (letters, circulars, payslips, task files).
- Ensures controlled access through role-based permissions.

Scalable & Integration Ready

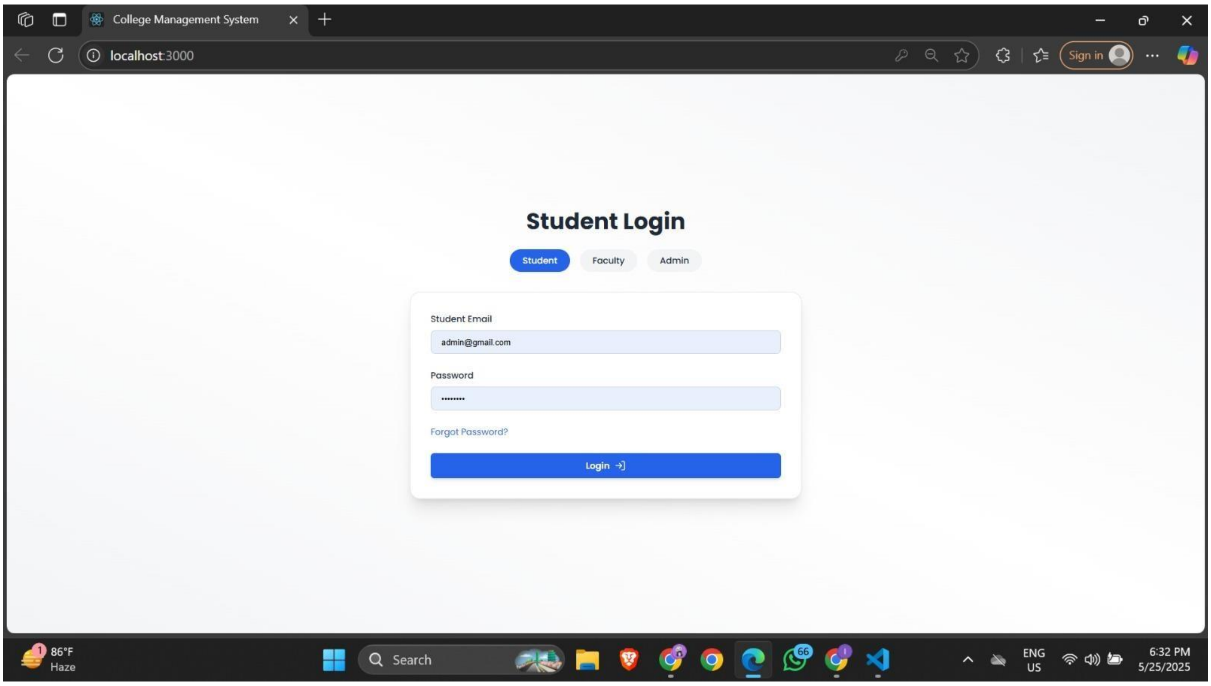
- Flexible architecture to add more modules like Exam Management, Student Module, Hostel, or Transport.
- Can integrate biometric machines or email notification systems through APIs.

7.2 User Interface Overview

The ERP Portal features a **modern and responsive UI** designed using React.js and styled components.

- The dashboard is personalized for each user role (Admin, Principal, Teacher, Coordinator).
- Side navigation includes all important modules visible based on role-based access.
- Interactive graphs and analytics provide real-time performance insights.
- The interface supports easy navigation even for non-technical users.

- Mobile-friendly design ensures access from any device with internet connectivity.



Student Login

Student Email

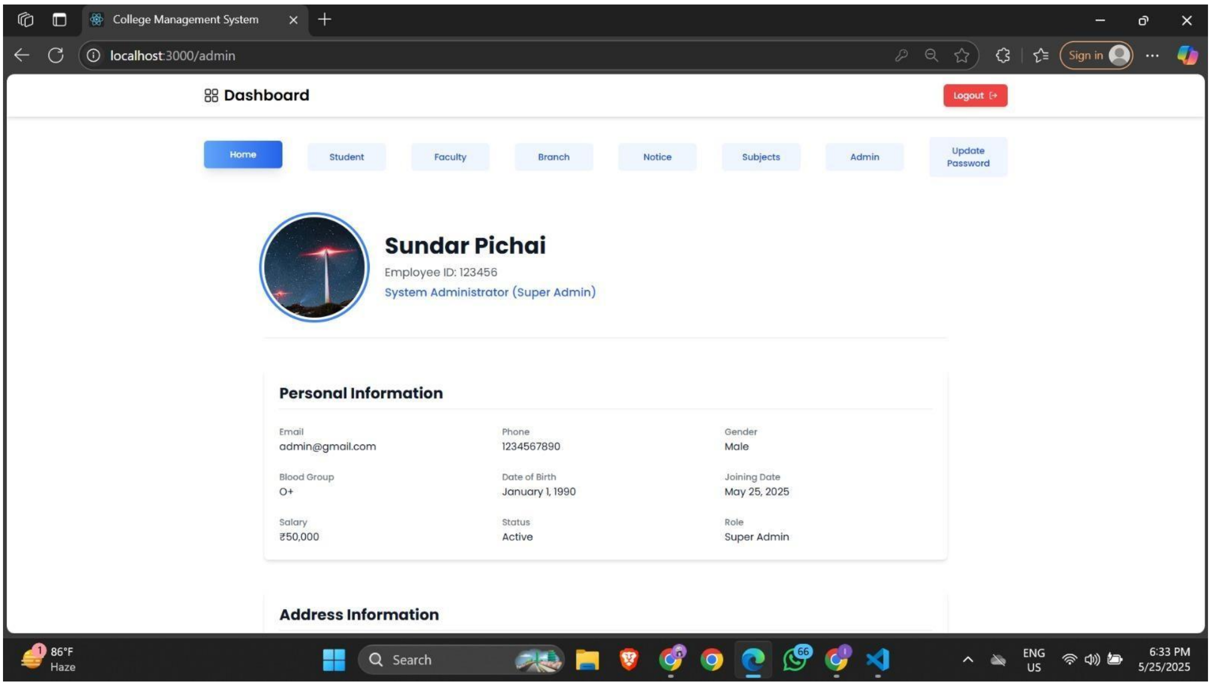
admin@gmail.com

Password

••••••••

Forgot Password?

Fig.3 Student Login



Sundar Pichai

Employee ID: 123456

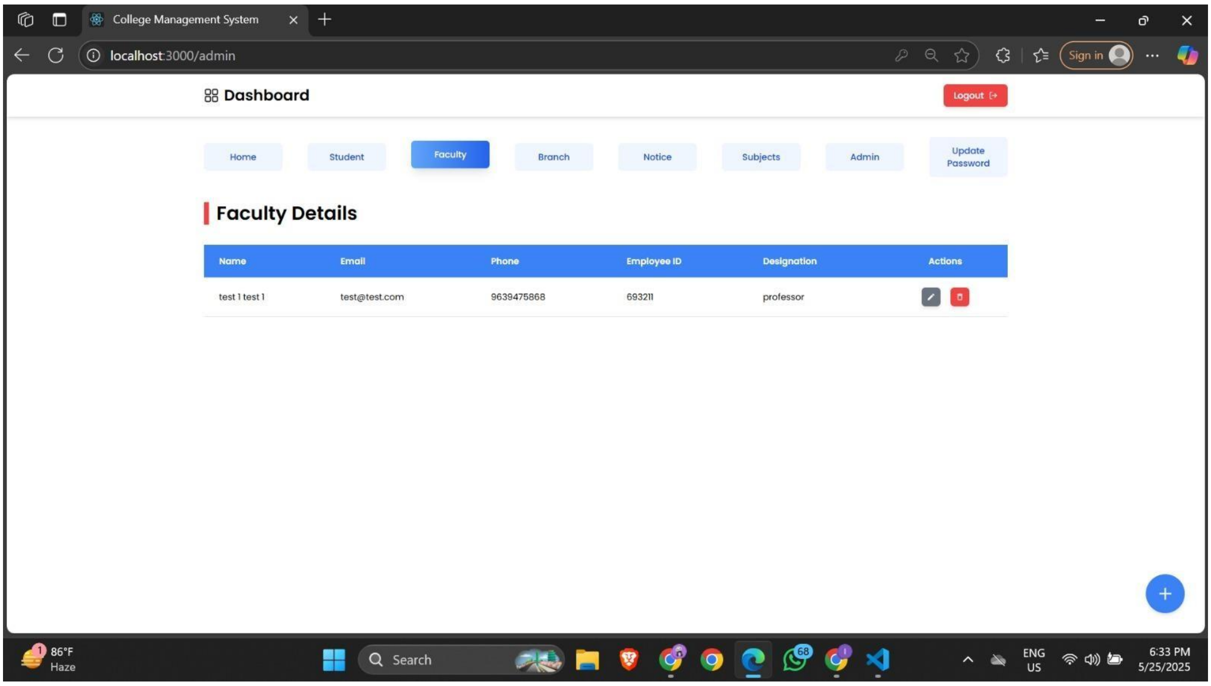
System Administrator (Super Admin)

Personal Information

Email admin@gmail.com	Phone 1234567890	Gender Male
Blood Group O+	Date of Birth January 1, 1990	Joining Date May 25, 2025
Salary ₹50,000	Status Active	Role Super Admin

Address Information

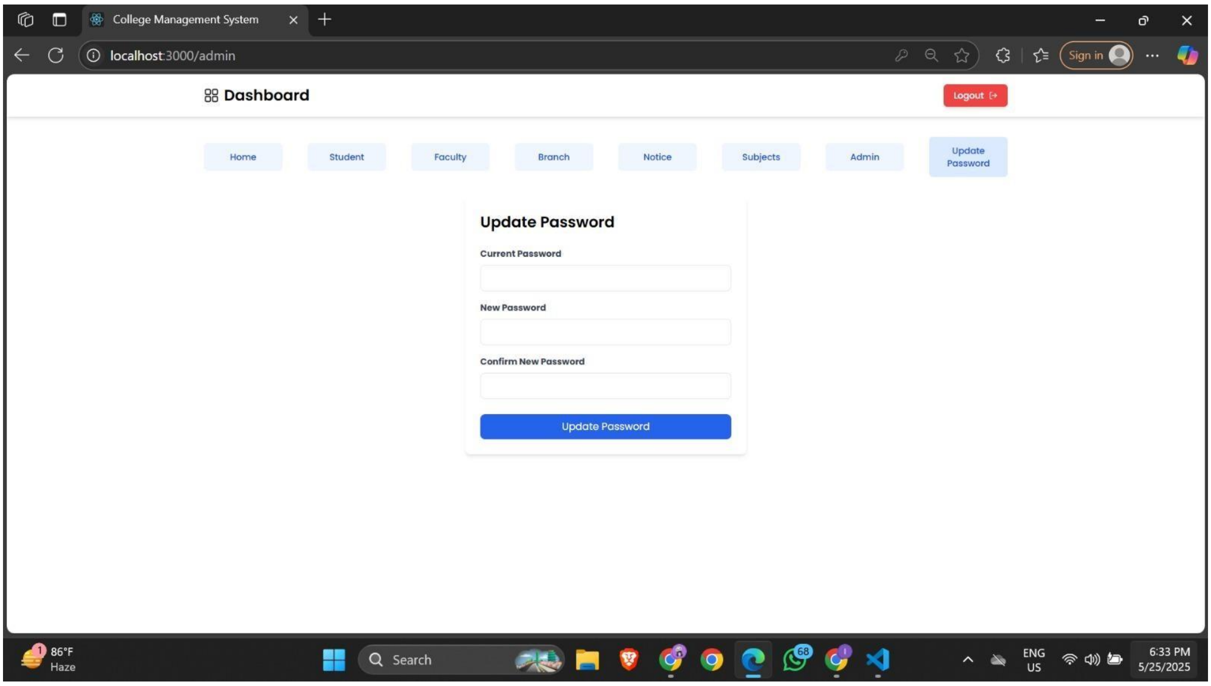
Fig.4 Dashboard



Faculty Details

Name	Email	Phone	Employee ID	Designation	Actions
test 1 test 1	test@test.com	9639475868	693211	professor	✎ 🗑

Fig.5 Faculty Details



Update Password

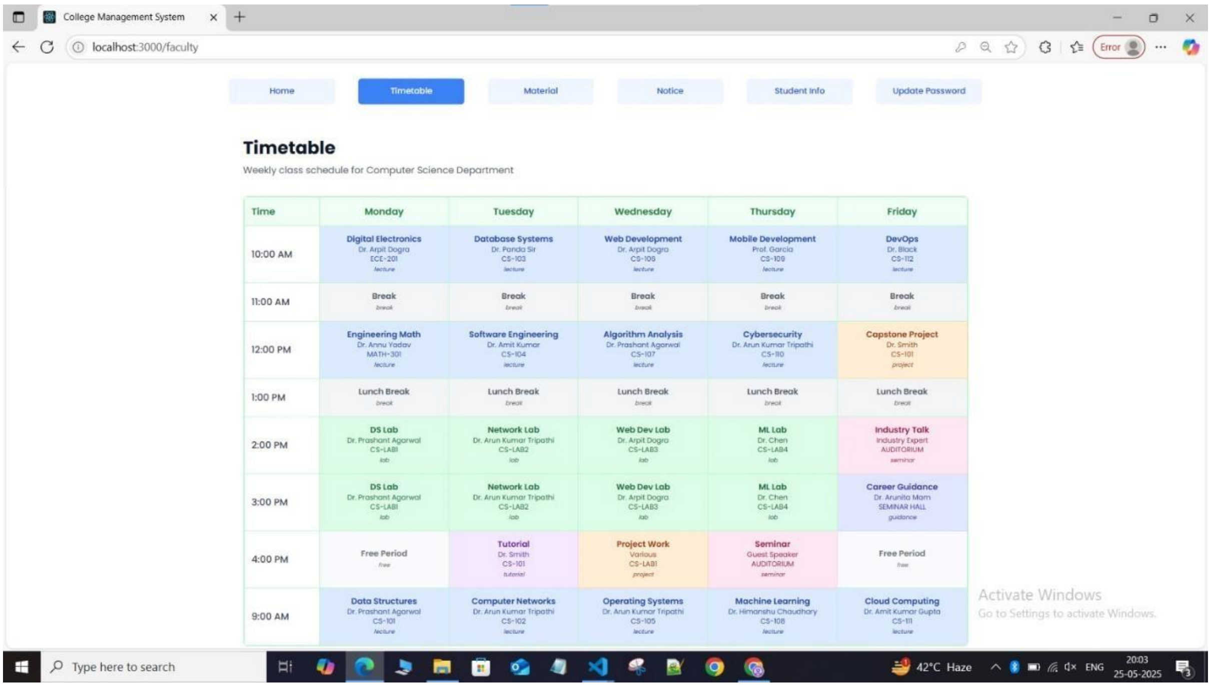
Current Password

New Password

Confirm New Password

Update Password

Fig.6 Update Password



Timetable

Weekly class schedule for Computer Science Department

Time	Monday	Tuesday	Wednesday	Thursday	Friday
10:00 AM	Digital Electronics Dr. Arpit Dogra ECE-201 *lecture*	Database Systems Dr. Panda Sir CS-103 *lecture*	Web Development Dr. Arpit Dogra CS-106 *lecture*	Mobile Development Prof. Garcia CS-109 *lecture*	DevOps Dr. Black CS-112 *lecture*
11:00 AM	Break *break*	Break *break*	Break *break*	Break *break*	Break *break*
12:00 PM	Engineering Math Dr. Annu Yadav MATH-301 *lecture*	Software Engineering Dr. Amit Kumar CS-104 *lecture*	Algorithm Analysis Dr. Prashant Agarwal CS-107 *lecture*	Cybersecurity Dr. Arun Kumar Tripathi CS-110 *lecture*	Capstone Project Dr. Smith CS-101 *project*
1:00 PM	Lunch Break *break*	Lunch Break *break*	Lunch Break *break*	Lunch Break *break*	Lunch Break *break*
2:00 PM	DS Lab Dr. Prashant Agarwal CS-LAB1 *lab*	Network Lab Dr. Arun Kumar Tripathi CS-LAB2 *lab*	Web Dev Lab Dr. Arpit Dogra CS-LAB3 *lab*	ML Lab Dr. Chen CS-LAB4 *lab*	Industry Talk Industry Expert AUDITORIUM *seminar*
3:00 PM	DS Lab Dr. Prashant Agarwal CS-LAB1 *lab*	Network Lab Dr. Arun Kumar Tripathi CS-LAB2 *lab*	Web Dev Lab Dr. Arpit Dogra CS-LAB3 *lab*	ML Lab Dr. Chen CS-LAB4 *lab*	Career Guidance Dr. Arunita Mam SEMINAR HALL *guidance*
4:00 PM	Free Period *free*	Tutorial Dr. Smith CS-101 *tutorial*	Project Work Various CS-LAB1 *project*	Seminar Guest Speaker AUDITORIUM *seminar*	Free Period *free*
9:00 AM	Data Structures Dr. Prashant Agarwal CS-101 *lecture*	Computer Networks Dr. Arun Kumar Tripathi CS-102 *lecture*	Operating Systems Dr. Arun Kumar Tripathi CS-105 *lecture*	Machine Learning Dr. Himanshu Chaudhary CS-108 *lecture*	Cloud Computing Dr. Amit Kumar Gupta CS-111 *lecture*

Fig.7 Timetable

REFERENCES

1. **MongoDB Documentation**

   https://www.mongodb.com/docs/

   Official documentation for MongoDB, used for designing the database schema and integrating CRUD operations for data storage.

2. **Express.js Documentation**

   https://expressjs.com/

   Reference for building backend APIs, middleware handling, routing, and server-side logic.

3. **React.js Documentation**

   https://react.dev/

   Used to develop the frontend user interface, handle component-based architecture, state management, and routing.

4. **Node.js Documentation**

   https://nodejs.org/en/docs/

   Provides guidance on implementing server-side logic, asynchronous programming, and REST API execution.

5. **JWT Authentication Guide**

   https://jwt.io/

   Reference used to implement secure authentication and authorization using JSON Web Tokens.

6. **Mongoose Documentation**

   https://mongoosejs.com/

   Used to define schemas and manage MongoDB operations from Node.js.

7. **Bootstrap Documentation**

   https://getbootstrap.com/docs/

   For responsive UI design and layout styling.

8. **Git and GitHub Documentation**

   https://docs.github.com/en/get-started/quickstart

   Used for version control, repository management, and project collaboration.

9. **Postman API Platform**

   https://www.postman.com/

   Used for API testing, debugging, and validating backend endpoints.

10. **Agile Software Development Principles**

Beck, K., et al. *"Manifesto for Agile Software Development."* Agile Alliance.

https://agilemanifesto.org/

Methodology followed during iterative development and sprint planning.

11. **W3Schools HTML, CSS, JavaScript Tutorials**

https://www.w3schools.com/

Reference for frontend fundamentals and UI styling practices.